

Assignment 01

1. Simple Car Controller

Simulate by using `Simple` and `Scene 3` in program `CarGUI`.

- `ZhichengSimpleCarControlTest.java`
- `ZhichengSimpleCarController.java`

2. Dubin Car Controller

Simulate by using `Dubin` and `Scene 3` in program `CarGUI`.

- `ZhichengDubinCarControlTest.java`
- `ZhichengDubinCarController.java`

3. Optimal Dubin Car Controller

Simulate by using `Dubin` and `Scene 3` in program `CarGUI`.

Definition and Simplification

1. The dubin car is simplified as a point and its shape is ignored. It means that "hit the obstacle" only happens when its coordinate crosses the boundary of the obstacle.
2. To counteract the influence of ignoring the shape, the obstacle is enlarged 10px on each side - the boundary is $\{(x, y) | 140 \leq x \leq 260, -10 \leq y \leq 110\}$.
3. The minimum time interval is 0.1 second.
4. Define variable $w1$ and $w2$ as the left and right speed of the dubin car in each phase, in order to distinguish from variable $v1$ and $v2$ (only in phase 2). Program `CarGUI` ONLY accepts the left and right speed in $[-10, 10]$.
 - In phase 1, $w1, w2$ are integers in $[-10, 10]$ and $w1 = -w2$ (rotate only - keep the coordinate of the dubin car unchanged).
 - In phase 2, $w1$ and $w2$ are integers in $[0, 10]$, which is the range of $v1$ and $v2$.
 - In phase 3, $w1, w2$ are rationals in $[-10, 10]$ (rotate the dubin car precisely to "facing right") and $w1 = -w2$.
 - In phase 4 (if exist), $w1 = w2 = 10$ (top speed).
5. The minimum interval of variable A is 0.05 radian, since it is the minimum rotate speed of the dubin car in phase 1. Also, A is no larger than 1.6 radian (91.6732°). To sum up $A = \{\frac{k}{20} | 0 \leq k \leq 32, k \in \mathbb{Z}\}$.
6. The minimum rotate speed of the dubin car in phase 1 is $+0.05$ radian - keep $w1 = -1$ and $w2 = 1$ for 1 tick.

7. The maximum rotate speed of the dubin car in phase 1 and phase 3 is ± 0.5 radian - keep $w_1 = \mp 10$ and $w_2 = \pm 10$ for 1 tick. Rotate more means more ticks. Rotation ticks are also included in total time.
8. The coordinate of target point is (500, 50). Define target area as $\{(x, y) | 495 \leq x \leq 500, 45 \leq y \leq 55\}$. "hit the target" is simplified as cross the boundary of the target area.

Method

Based on the range of A, v_1, v_2 below, there are $32 \times 11 \times 11 = 3872$ possible input.

Use library `DubinCarSimulator.class` to simulate the driving process to simulate each input. During the simulation, the time of each step is set as 0.1 second.

After each step, check out whether the location of the dubin car violate restrictions below. If true, stop simulate and mark the input as invalid.

Keep doing this until all input has been handled. The optimal solution is the valid input with minimum time.

Restriction

1. $v_1 > v_2 \geq 0$ (keep turning right in phase 2)
2. The dubin car should NOT hit the obstacle.
3. The dubin car should ALWAYS in area $\{(x, y) | 50 \leq x \leq 500, 45 \leq y \leq 475\}$ (the upper border is about $y = 475$).

Realization

- `ZhichengStep.java`
- `ZhichengSolution.java`
- `ZhichengOptimal.java`
- `ZhichengOptimalDubinCarController.java` - based on the optimal solution

Result

Parameters of Optimal Solution:

$A = 0.9\text{rad}$
 $v_1 = 10$
 $v_2 = 9$

Steps of Optimal Solution: (10600ms)

```
[{ left: -10.0, right: 10.0, time: 100, globalTime: 100 }  
, { left: -8.0, right: 8.0, time: 100, globalTime: 200 }  
, { left: 10.0, right: 9.0, time: 100, globalTime: 300 }  
, { left: 10.0, right: 9.0, time: 100, globalTime: 400 }  
, { left: 10.0, right: 9.0, time: 100, globalTime: 500 }]
```

[illegible]

[illegible]

```
, { left: 10.0, right: 10.0, time: 100, globalTime: 9700 }  
, { left: 10.0, right: 10.0, time: 100, globalTime: 9800 }  
, { left: 10.0, right: 10.0, time: 100, globalTime: 9900 }  
, { left: 10.0, right: 10.0, time: 100, globalTime: 10000 }  
, { left: 10.0, right: 10.0, time: 100, globalTime: 10100 }  
, { left: 10.0, right: 10.0, time: 100, globalTime: 10200 }  
, { left: 10.0, right: 10.0, time: 100, globalTime: 10300 }  
, { left: 10.0, right: 10.0, time: 100, globalTime: 10400 }  
, { left: 10.0, right: 10.0, time: 100, globalTime: 10500 }  
, { left: 10.0, right: 10.0, time: 100, globalTime: 10600 }  
]
```